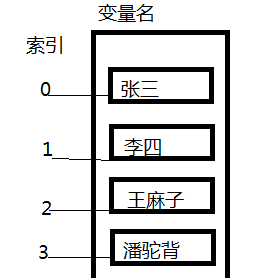
1. 数组
   1. 什么是数组

**数组**是一组数据有序排列的集合。将一组数据按一定顺序组织为一个组合，并对这个组合命名，这样便构成了数组。



**数组元素**：组成数组的每一个数据称为数组的一个数组元素。

**数组索引**：每一个数组元素对应一个整数值，称为数组元素索引，或者数组元素下标。元素索引为非负整数，由0开始依次增加，即第一个元素索引为0，依次为1，2等。

**数组元素的访问**：对数组中各个元素，使用数组名加上以方括号”[ ]”括起来的元素索引进行访问。元素索引为元素在数组中的位置序号。

**数组的长度**：数组元素的个数称为数组的长度。

数组是复合数据类型，都可以保存多个数据

语法格式：

var arr1 = new Array();//定义一个空数组

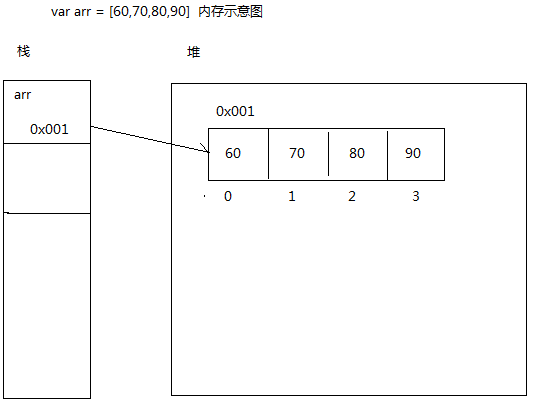
var arr2 = new Array(1,25,37,46,95);//定义了一个具有5个元素的数组

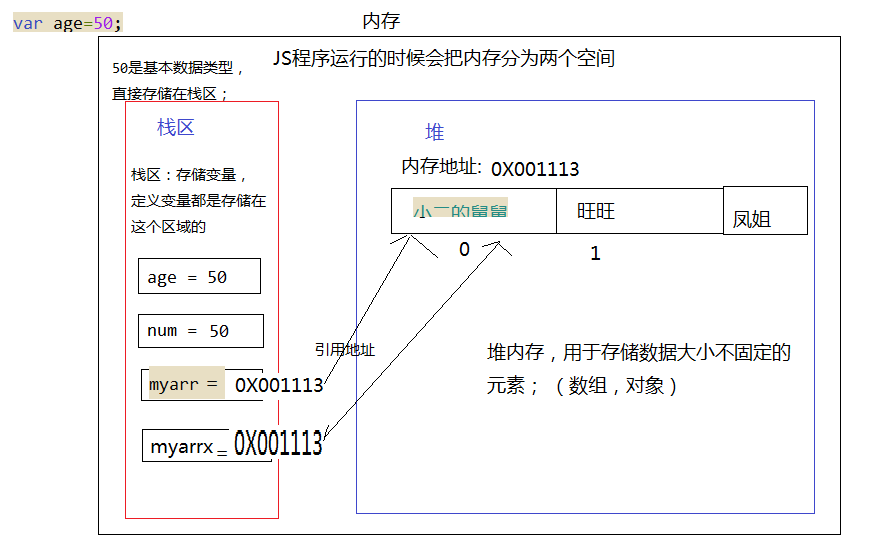
console.log(arr2);

这种写法较简练，并且使用很普遍

var arr3 = [1,3,5,7,9];//创建了一个带有5个元素的数组

* 1. 内存分布





* 1. 什么时候使用到数组

需要将多个类型一样的数据一次性告诉计算机让其保存时使用数组类型。

例如：

告知计算机保存一个班的成绩

告知计算机保存一个班的年龄，名字。

* 1. 访问数组中的元素

语法：

数组变量名[索引值]

例如：arr2[1]

console.log(arr2[1]);

通过下标获取元素值，如果没有定义这个元素，则返回undefined

var arr = new Array(1,25,37,46,95);//定义了一个具有5个元素的数组

比如获取arr中第2个元素的值：arr2[1];

arr[5];//返回undefined，因为arr最大下标为4，并没有5，所以会返回undefined

* 1. 访问数组长度

语法：

数组变量名.length;

例如：

var len = arr.length;

* 1. 删除数据中的元素

语法：

delete 数组变量名 [索引]

delete arr[2];

使用delete删除数组元素与为其赋值为undefined类似。

使用delete删除元素并不会修改length属性

清空数组：

arr = [];

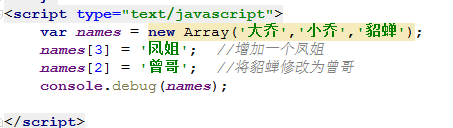
或

arr.length=0;

* 1. 修改和增加元素

语法：

数组变量名称[索引] = 值



* 1. 数组的遍历
     1. 什么是数组的遍历

所谓数组的遍历就是挨个取出数组中每个下标对应的每一项。

* + 1. 如何遍历出数组元素

**for语法结构：**

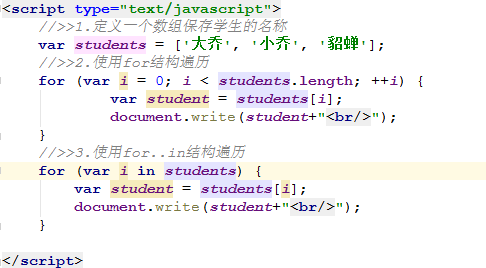
**for**(**var *i*** = 0; ***i***< 数组变量.**length**;++***i***){  
 **var *元素变量*** = 数组变量[***i***];  
}

**for…in语法结构：**

**for**(**var *i* in** 数组变量){  
 **var *元素变量*** = 数组变量[***i***];  
}

其中i变量可以是任何变量的名字，作为数组索引的功能。

例如：



* + 1. 课堂练习
       - 练习一

数组为：[1,3,5,7,24,32,25,48]

1.求数组奇数的个数和偶数的个数

2.求出上面数组的平均值

* + - * 练习二

设计一个数组中的最大值的功能

[1,3,5,79,11]

[-3,-5,-7,-121];

# 二维数组

* 1. 什么是二维数组

二维数组本质上是以数组作为数组元素的数组。

var arr1 = [1,2,3,4];

var arr2 = [5,6,7,8];

var arr3 = [9,10,11,12];

以上的三个数组中的元素都是普通的值,此类数组被称为一维数组.

var arr4 = [arr1,arr2,arr3];

arr4数组中的元素也是数组,arr4变量的数组被称为二维数组. 如果将arr4变量作为另外数组的元素,另外数组就叫做三维数组,依次类推.

也可以写成一下格式:

var arr4 = [

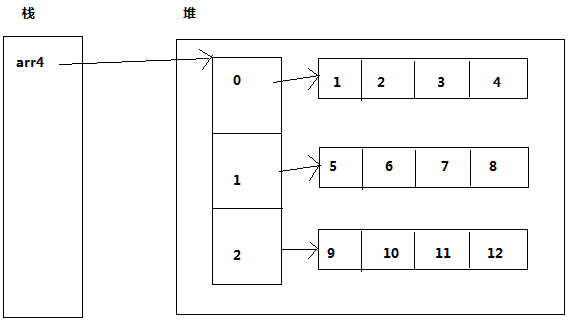
[1,2,3,4],

[5,6,7,8],

[9,10,11,12]

];

该数组在内存中的分布图如下:



* 1. 二维数组在哪里使用

数组就是有限个元素按一定顺序排列的集合,如果该数据集合又处于另外一个数据集合中,那么就可以使用二维数组来表示.

例如,一个小组的学生成绩:

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 语文 | 数学 | 英语 |
| 张 | 80 | 92 | 92 |
| 王 | 61 | 65 | 71 |
| 李 | 59 | 63 | 70 |
| 赵 | 85 | 87 | 90 |
| 周 | 76 | 77 | 85 |

每个学生的成绩使用一维数组表示,当一个小组的学生成绩必须再使用一个数组存放每个学生成绩的数组.如下:

学生成绩:

var zhang = [80,92,92];

var wang = [61,65,71];

var li = [59,63,70];

var zhao = [85,87,90];

var zhou = [76,77,85];

小组成绩:

var group = [

zhang,

wang,

li,

zhao,

zhou

]

或者:

var group = [

[80,92,92],

[61,65,71],

[59,63,70],

[85,87,90],

[76,77,85]

];

再举例: 如将每个班级看成学生集合, 那么每个校区就是一个二维数组. 整个就一个三维数组(数组中放数组,数组中再存放数组.)

* 1. 常见二维数组的使用方法
     1. 访问二维数组中的具体元素

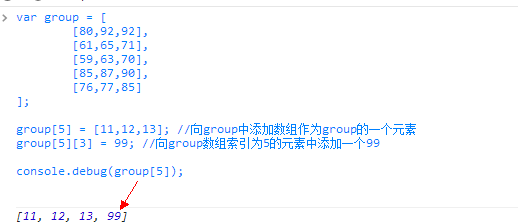


group[x] : 访问到二维数组中索引为x数组类型的元素

group[x][y]: 访问到二维数组中索引为x数组类型的元素,然后再通过y访问到数组类型元素中具体一个元素.

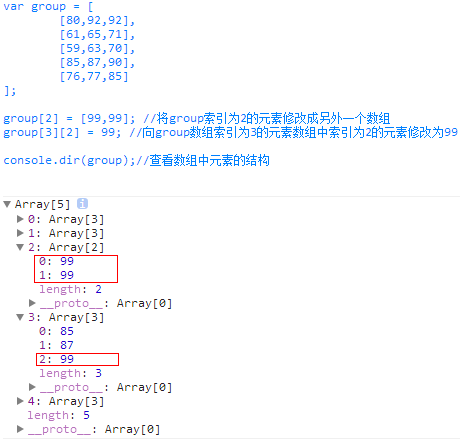
* + 1. 向二维数组中添加元素

通过新的索引为数组中添加具体的元素

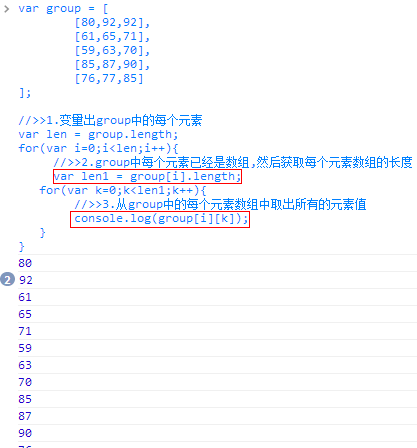


* + 1. 修改数组中的元素

通过已有索引修改数组中的元素



* + 1. 二维数组的遍历



* 1. 二维数组实例

找出以下数组中最大值及最大值对应的索引值:

var group = [

[80,92,92],

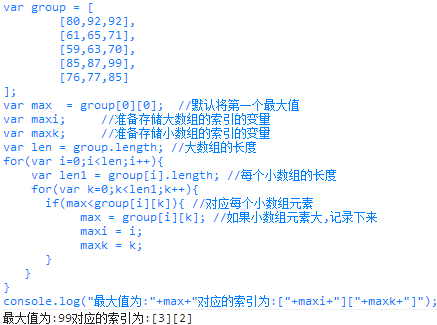
[61,65,71],

[59,99,70],

[85,87,90],

[76,77,88]

];



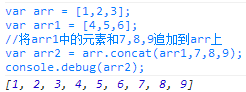
* 1. 数组自身常用方法

数组自身函数是系统已经为数组定义好的方法,可以通过数组直接调用的方法.以下函数必须通过数组调用执行.

* + 1. arrayObject.concat(arrayX,arrayX,......,arrayX)

|  |  |
| --- | --- |
| arrayX | 必需。该参数可以是具体的值，也可以是数组对象。可以是任意多个。 |

连接两个或更多的数组，并返回结果。



* + 1. arrayObject.join(separator)

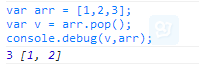
|  |  |
| --- | --- |
| separator | 可选。指定要使用的分隔符。如果省略该参数，则使用逗号作为分隔符。 |

把数组的所有元素放入一个字符串。元素通过指定的分隔符进行分隔。



* + 1. arrayObject.pop()

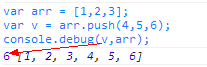
删除并返回数组的最后一个元素



* + 1. arrayObject.push(newelement1,[newelement2]….)

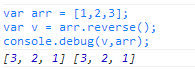
|  |  |
| --- | --- |
| newelement1 | 必需。要添加到数组的第一个元素。 |
| newelement2 | 可选。要添加到数组的第二个元素。 |
| …. | 可选。可添加多个元素。 |

向数组的末尾添加一个或更多元素，并返回新的长度。



* + 1. arrayObject.reverse()

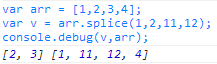
颠倒数组中元素的顺序。 但该方法不会创建新的数组



* + 1. arrayObject.splice(index,howmany,element1,.....,elementX)

|  |  |
| --- | --- |
| index | 必需。规定从何处添加/删除元素。  该参数是开始插入和（或）删除的数组元素的下标，必须是数字。 |
| howmany | 必需。规定应该删除多少元素。必须是数字，但可以是 "0"。  如果未规定此参数，则删除从 index 开始到原数组结尾的所有元素。 |
| element1 | 可选。规定要添加到数组的新元素。从 index 所指的下标处开始插入。 |
| elementX | 可选。可向数组添加若干元素。 |

删除元素，并向数组添加新元素。

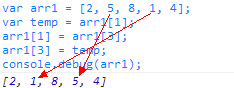


* 1. 交换数组中两个索引上的值

首先解决交换两个变量之间的值:  


var arr1 = [2, 5, 8, 1, 4];

交换索引 1和3的值:



* 1. 数组排序

数组排序一般是指将数字数组中的元素按照从大到小或者是从小到大的顺序排列.

var arr1 = [2, 5, 8, 1, 4];

排序后:

var arr2 = [1, 2, 4, 5, 8] 或者

var arr3 = [8, 5, 4, 2, 1]

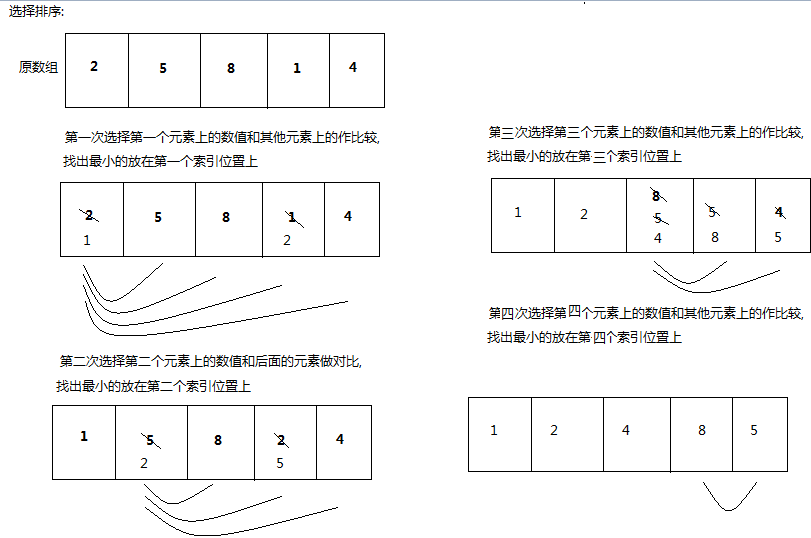
常见的排序算法有两种:

“选择排序”和”冒泡排序”

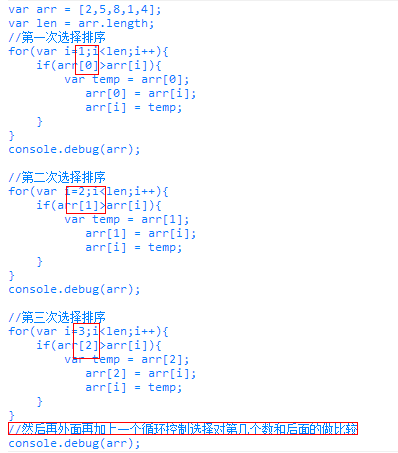
* + 1. 选择排序

选择排序(Selection sort)是一种简单直观的[排序算法](http://zh.wikipedia.org/wiki/%E6%8E%92%E5%BA%8F%E7%AE%97%E6%B3%95" \o "排序算法)。它的工作原理如下。首先在未排序序列中找到最小（大）元素，存放到排序序列的起始位置，然后，再从剩余未排序元素中继续寻找最小（大）元素，然后放到已排序序列的末尾。以此类推，直到所有元素均排序完毕。

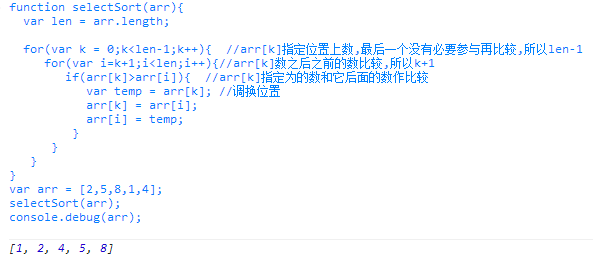
图解排序过程:



通过程序模拟排序过程:



最终排序算法:

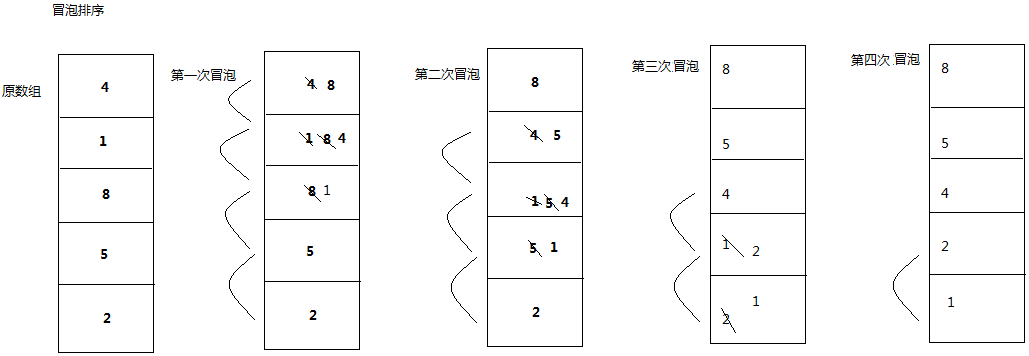


如果要从大到小排: 只需要改变比较规则即可.将>改成<.

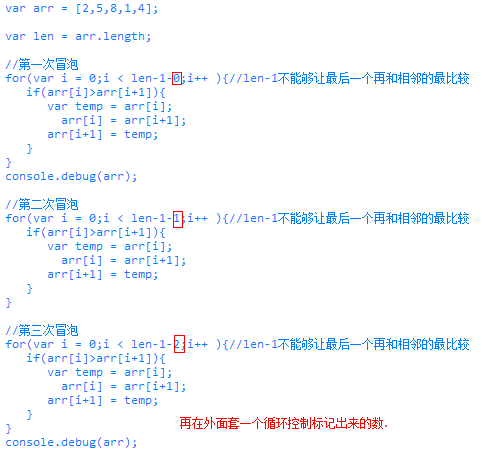
* + 1. 冒泡排序

冒泡排序是它重复地走访过要排序的数列，一次比较相邻的两个元素，如果他们的顺序错误就把他们交换过来。走访数列的工作是重复地进行直到没有再需要交换，也就是说该数列已经排图解排序过程:

图解排序过程:



通过程序模拟排序过程:



完成冒泡排序的算法:

